

TheComputingTutor ©2021

Maths Quiz

Program Notes for BTEC IT Unit 4

Learning Aim A – Code Analysis



Table of Contents

Overview - What is the program?	3
Getting Started – The Text File.....	4
The Main Menu	6
Taking a Quiz – Username and Difficulty.....	Error! Bookmark not defined.
Taking a Quiz – Answering The Questions.....	Error! Bookmark not defined.
Quiz Statistics Submenu	Error! Bookmark not defined.
Total Quiz Data.....	Error! Bookmark not defined.
Quiz Difficulty Data.....	Error! Bookmark not defined.
Displaying The User Quiz Data	Error! Bookmark not defined.
Displaying the Question Data	Error! Bookmark not defined.
Error Log File	Error! Bookmark not defined.



Overview - What is the program?

The program is meant to simulate a Maths Quiz to be used by primary school or Year 7 maths students. The idea is that there are three sets of questions: easy, medium and difficult.

Each row of these files contains a simple maths question.

The list of questions is loaded into the program, which has a set of difficulty levels built in. As students progress through the program, a new quiz is created. If the student advances in difficulty, the mix of easy, medium and difficult questions changes. The Easy level 1 has only easy questions, the difficult level 5 has only hard questions.

At the end of each quiz, the results are saved to a new file. Students can view statistics on quiz operations.

The program makes extensive use of:

- User Input validation
- Lists and Dictionaries
- Try Catch Finally blocks
- While loops
- For loops
- File Input and Output.

The program code has been especially designed to use only the code techniques covered in the syllabus. It is not meant to represent a definitive maths quiz, however it will provide enough of a detailed code base for students to analyse for Learning Aim A.



Getting Started – The Text File

Key information is held in the files EasyQuestions.txt, MediumQuestions.txt and DifficultQuestions.txt

The EasyQuestions file contains the following information:

```
E1,1,*,2,2
E2,2,*,2,4
E3,2,*,3,6
E4,2,+,2,4
E5,2,+,3,5
E6,2,-,1,1
E7,5,-,2,3
E8,8,/,2,4
E9,10,/,5,2
```

Figure 1.1 – the EasyQuestions text file data for the program.

```
M1,8,*,5,40
M2,12,*,12,144
M3,9,*,7,63
M4,21,+,12,33
M5,17,+,9,26
M6,21,-,7,14
M7,35,-,13,22
```

Figure 1.2 – the MediumQuestions text file data for the program.

```
D1,18,*,51,900
D2,122,*,9,1098
D3,91,*,77,7007
D4,21,+,12,33
D5,85,+,123,208
D6,255,-,63,192
D7,423,-,387,36
```

Figure 1.3 – the DifficultQuestions text file data for the program.

Each row of data contains the following information:

- A question ID starting with E, M and D for Easy Medium and Difficulty
- A number
- A mathematical operator
- Another number
- The correct answer to the question



Quiz data is logged in the file QuizData.txt. The information contained in this file is shown in figure 1.4:

```
NAME,DIFFICULTY,QUIZ,RIGHT,WRONG,SCORE
test,1,E1-E2-E3-E4-E5,E1-E2,E3-E4-E5,2
jon,1,E6-E1-E8-E3-E4,E6-E1-E8-E3-E4,NONE,5
jon,2,E5-E1-E8-M6-M1,E5-E1-E8-M6-M1,NONE,10
jon,3,E6-M5-M2-M7-D2,E6-M5-M2-M7-D2,NONE,15
fred,1,E7-E2-E3-E6-E1,E7-E2-E3-E6,E1,4
jon,1,E6-E4-E7-E3-E5,E6-E4-E7-E3-E5,NONE,5
```

Figure 1.4 – the Quiz Data text file data for the program.

The first row is a header row and identifies the following columns:

- NAME – the name of the student taking the quiz
- DIFFICULTY – the level of difficulty of the quiz
- QUIZ – a list of ID's of the questions in that randomly generated quiz
- RIGHT – the ID's of the questions the student got correct. If none are correct, the word NONE is inserted
- WRONG – the ID's of the questions the student got wrong. If none are wrong, the word NONE is inserted
- SCORE – the score the student obtained for the quiz.

The items are separated by a ',' to allow the line to be split into a string array.

The sub-lists QUIZ, RIGHT and WRONG are separated by a '-' to allow the data to be split into a string array.

Note the last item on the list has no ',' after it and this logic must be dealt with in code.



The Main Menu

On running the program, the Main Menu is displayed:

```
Welcome To TheComputingTutor Maths Tutorial Quiz

Please choose from the following Menu Options
1: would you like to take a Quiz?
2: would you like to view Quiz statistics
Please enter 1 or 2: This is not a valid number please try again

Please choose from the following Menu Options
1: would you like to take a Quiz?
2: would you like to view Quiz statistics
Please enter 1 or 2: _
```

Figure 2 the main menu of the program.

The program then displays the standard error handling.

Any user input must be of the correct datatype (string, int) and must be a valid number.

This is carried out by a combination of IF ELIF statements to check for validity, a WHILE loop to prompt the user to have another go, and a TRY EXCEPT statement to check for the correct datatypes being entered.



```
Welcome To TheComputingTutor Maths Tutorial Quiz
```

```
Please choose from the following Menu Options
```

```
1: would you like to take a Quiz?
```

```
2: would you like to view Quiz statistics
```

```
Please enter 1 or 2: fred
```

```
This is not a valid option please try again
```

```
Please choose menu option 1 or 2: 5
```

```
This is not a valid option please try again
```

```
Please choose menu option 1 or 2:
```

Figure 3 – user input validation for the main menu.

If a user enters a string, when the program tries to convert this to a number a Exception is caught.

When this happens, the except block sets a variable to -1 and returns this number. If the calling code detects this number, then the user has entered the wrong datatype and the program will prompt the user for another go.

Key Functions	
Function Name	Purpose
RunProgram()	The main entry point to the program. This calls BuildQuestionList and DisplayMainMenu()
BuildQuestionList()	This loads each of the three question files into it's own array for processing within the program
FileLoad()	A function that loads a specified question file into an array. It will then build a dictionary of each question and load that into a list. It will then add it to the specified list.
LogError()	A utility function that logs errors to a file.
DisplayMainMenu()	Displays the main menu to the screen.
checkNumericalInput()	A function that makes sure the user input is a number. Returns -1 to the calling code if the input is a string.
TakeAQuiz()	Function to take a quiz.
DisplayStatisticsMenu()	Function to show the statistics menu





No part of this publication may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the TheComputingTutor

